Argonne National Laboratory

MVT Accounting Routine for Release 17 of OS/360

by

David R. Snider

The facilities of Argonne National Laboratory are owned by the United States Government. Under the terms of a contract (W-31-109-Eng-38) between the U. S. Atomic Energy Commission, Argonne Universities Association and The University of Chicago, the University employs the staff and operates the Laboratory in accordance with policies and programs formulated, approved and reviewed by the Association.

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

Printed in the United States of America
Available from
Clearinghouse for Federal Scientific and Technical Information
National Bureau of Standards, U. S. Department of Commerce
Springfield, Virginia 22151
Price: Printed Copy $3.00; Microfiche $0.65

ANL-7707
Mathematics and Computers

ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

MVT Accounting Routine for Release 17 of OS/360

by

David R. Snider

Applied Mathematics Division

July 1970

TABLE OF CONTENTS

MVT ACCOUNTING ROUTINE FOR RELEASE 17 of OS/360

by

David R. Snider

ABSTRACT

This is a description of the Accounting Routine used on the System/360 Model 75 running under Multiprogramming with a Variable Number of Tasks (MVT), Release 17, at Argonne National Laboratory. It includes the definitions of all statistics gathered, a description of how to apply the routine to the system, and listing of the Accounting Routine and all system modifications.

I. INTRODUCTION

OS/360 includes provisions for a user-written accounting routine. The CSECT name for this routine is IEFACTRT. The *IBM System/360 Operating System Programmer's Guide*[1] describes the data passed by OS to an accounting routine and how such a routine is implemented. This information is essential for a full understanding of this report.

Argonne's Model 75 is supported by a Model 50 running under the Attached Support Processor (ASP). The two machines are connected by a Channel-to-Channel Adapter. The accounting routine passes information to ASP via Write-to-Operator instructions. ASP puts the user information in each job's SYSMSG file, and uses the amount of main core storage and Large Capacity Storage (LCS) requested and the CPU and Voluntary Wait (I/O) times (defined in Section IV) to calculate the number of charge units for the job. ASP also uses the time estimates (defined in Section V), along with other information, to schedule jobs to be run on the 75.

It was apparent, when MVT was first implemented at Argonne, that more than the information made available by OS would be required. Information was needed by ASP to calculate the charge for a job. The users needed to know the amounts of Main Core Storage and LCS they were actually using, and the amounts of CPU and I/O time taken to execute each step and job. Also, the Computer Center needed information to analyze system efficiency and jobstream attributes. The scope of the user accounting routine was therefore extended beyond that provided by IBM.

II. MESSAGES

The accounting routine's main function is to calculate and record statistics. The routine uses four messages to pass statistics from each job and job step to the support processor and to the SYS1.ACCT data set on the main processor IPL pack.

Message ACCT004 is issued at step initiation. It is used to record the job name, step name, and initiation time on the main processor of each job step. Since this message contains no user information, it is written only on SYS1.ACCT.

Messages ACCT002 and ACCT003 contain most of the user statistics and are sent to both the support processor and SYS1.ACCT. Both these messages are issued at step termination.

Message ACCT001 is issued at job termination. This message records total CPU and WAIT times for the job and is sent to the support processor and to SYS1.ACCT.

Examples of the messages are shown in Fig. 1. The first group of messages shows what appears in the SYSMSG file of a user's output. The second group shows the way the messages are written into the SYS1.ACCT data set. The messages have a time stamp added to the end to indicate the exact time that they are written on SYS1.ACCT.

Each message begins with the message number and the job name for which it was issued. This is followed, for all messages except ACCT001, by the step name. The format of the remainder of each message is described below. The meaning of each statistic and the method used to calculate it are explained in the Section IV, "Definition of Statistics." All numbers are decimal unless specified otherwise.

Message ACCT002 contains the CPU and Voluntary Wait times in minutes, seconds, and hundredths of a second. Up to five digits are printed for minutes; any job longer than that will have the high order digits truncated.

RQST in ACCT002 stands for Requested Regions. The regions requested in hierarchy 0 (Main Core Storage) and in hierarchy 1 (Large Capacity Storage) are given in number of K (1024 bytes) of storage.

The end of message ACCT002 has a twelve-digit field of hexadecimal numbers. The first four digits represent the CPU time for the step in hundredths of a second. The fifth through the eighth digits represent the Voluntary Wait time in hundredths of a second. The ninth and tenth digits contain the number of K requested in main core. The eleventh and twelfth digits contain the number of K requested in LCS. This information is used by ASP to calculate the number of charge units for a job.

SIO, EXCP, and WAIT are the number of start I/O instructions issued, the number of SVC 0's, and the number of SVC 1's, respectively. These numbers are allowed a maximum of five digits each; anything more than that is truncated at the high end.

USED in ACCT003 stands for the actual amount of core used in each region. H0 and H1 used are represented in number of K.

Message ACCT001 contains the TOTAL CPU and TOTAL WAIT times for this job. These totals should be equal to the sum of the CPU and the Voluntary Wait times for all the steps. This message also contains the recommended time estimate for the job.

```
R=ACCT002 LIB01172/ASM      CPU      .  5.40 WAIT      . 14.44 RQST H0 200K,H1    OK
R=ACCT003 LIB01172/ASM      SIO  1439 EXCP   934 WAIT   741    USED H0 200K,H1    OK

R=ACCT002 LIB01172/LINK     CPU      .  1.07 WAIT      . 12.01 RQST H0 200K,H1    OK
R=ACCT003 LIB01172/LINK     SIO   8.5 EXCP   563 WAIT   535    USED H0 130K,H1    OK

R=ACCT002 LIB01172/LINK     CPU      .  1.02 WAIT      . 14.16 RQST H0 200K,H1    OK
R=ACCT003 LIB01172/LINK     SIO   895 EXCP   652 WAIT   621    USED H0 130K,H1    OK
R=ACCT001 LIB01172 TOTAL CPU      .  7.49 TOTAL WAIT      . 40.61 TIME EST      1
```

(a)

```
ACCT004 LIB01172/ASM          STEP INITIATED                                         7001109184888

ACCT002 LIB01172/ASM        CPU      .  5.50 WAIT      . 14.64 RQST H0 200K,H1    OK 7001109191657

ACCT003 LIB01172/ASM        SIO  1451 EXCP   934 WAIT   757    USED H0 200K,H1    OK 7001109191657

ACCT004 LIB01172/LINK         STEP INITIATED                                         7001109191706

ACCT002 LIB01172/LINK       CPU      .  1.13 WAIT      . 20.56 RQST H0 200K,H1    OK 7001109194302

ACCT003 LIB01172/LINK       SIO  1088 EXCP   606 WAIT   575    USED H0 130K,H1    OK 7001109194302

ACCT004 LIB01172/LINK         STEP INITIATED                                         7001109194339

ACCT002 LIB01172/LINK       CPU      .  1.17 WAIT      . 22.40 RQST H0 200K,H1    OK 7001109201267

ACCT003 LIB01172/LINK       SIO  1182 EXCP   659 WAIT   629    USED H0 130K,H1    OK 7001109201267

ACCT001 LIB01172 TOTAL CPU        .  7.80 TOTAL WAIT      . 57.60 TIME EST      2     7001109201284
```

(b)

Fig. 1. Typical accounting messages as they appear in (a) SYSMSG, (b) SYS1.ACCT

III. SYSTEM REQUIREMENTS

The accounting routine requires about 2000 bytes of core storage each time it is loaded. Because the routine is part of each Initiator, the amount of core required by each Initiator to execute is increased. Each Initiator uses 212 bytes of System Queue Space for a register save area and for space to build messages. This SQS is freed before the job-step gets control and, of course, all the dynamic area core used by each Initiator is freed also.

A work area of 40 bytes is needed for each Initiator. The Initiator gets core for the work area from SQS at job initiation and keeps it until job termination. The work area is used to hold the statistics for the job step as they are gathered. Its format is

0	SIO Count	EXCP Count	
8	Wait Count	Volunary Wait Time for Step	
16	Actual Core Used	Actual LCS Used	
24	Voluntary Wait Time for Job	Modified Wait Time for Step	
32	Modified Wait Time for Job	Cell EXCP's Pending	Disk EXCP's Pending

Several parts of the Nucleus were modified in order to gather statistics. The Getmain routine (IEAQGM00) was modified to calculate minimum free space, which is used to find actual core and LCS used. The Trace routine (IEAQTRCE) was changed in two places, the first to count SIO's and the second to count EXCP's. The Post Routine (IGC002) was modified to count waits, keep track of voluntary wait time, and add this time to CPU time for time limits. The Time Left routine requires that the TTIMER routine (IGC046) be slightly modified. All these changes require only 372 bytes to be added to the size of the Nucleus.

Auxiliary storage is required for SYS1.ACCT. The DCB for SYS1.ACCT is (RECFM=F,LRECL=96,BLKSIZE=96). One 2314 cylinder can accommodate 720 blocks of 96-bytes. SYS1.ACCT was allocated 15 cylinders on SYSRES, and will hold statistics for 1080 compile-load-go jobs. The contents of SYS1.ACCT are dumped and the data set is scratched and reallocated once per day, using IEBPTPCH and IEHPROGM.

IV. DEFINITION OF STATISTICS

The methods used for finding values of the statistics need explanation since several of them allow a number of different possible interpretations. The method of calculating each value is described below.

1. CPU Time

CPU time is the sum of the time intervals during which the step controls the CPU. This means that whenever the job step TCB or a sub-task TCB is the current TCB in the system (the one in control) the time is charged to that step.

OS/360 makes CPU time for each step available at step termination and CPU time for each job available at job termination, if the job step timer option is taken at System Generation.[1,2]

OS calculates the CPU time for a step by taking the time estimate for the step from the Step Control Table at step initiation and building a Timer Queue Element (TQE) with that value by issuing a STIMER (SVC 47). The value in the TQE is decremented for each interval during which the step is not in Wait state. At step termination the value in the TQE is subtracted from the value in the Step Control Table to yield the CPU time for the step.

2. Voluntary Wait Time

A job step is considered to be in a voluntary wait state between the time the entire step goes into the wait state and the time it again becomes dispatchable. In the simplest instance, a job step with no subtasks, this is the time between the issuance of a Wait SVC and the posting of the Event Control Block. This method allows charging a user for the time he spends doing I/O without charging him for the time he spends waiting for another job while he is dispatchable.

Wait time is calculated with a modification to the POST Routine (SVC 2). Each time the entire step goes into a wait, the value in the TQE is saved and a number representing the time limit for waits in timer units is put in its place. When the wait is posted and the step becomes dispatchable again, the length of the wait is added to a counter in the accounting work area for the step. The counter is for step wait time and is zeroed at step initiation.

Each voluntary wait currently has a time limit of thirty minutes, after which the job is abnormally terminated. If it becomes desirable, this time limit can be changed.

3. Requested Region Sizes

Every job step run on the main processor requests, either explicitly or implicitly, a certain amount of Main Core Storage and a certain amount of Large Capacity Storage. The values requested are stored in the Step Control Table for each job step. The accounting routine copies the values from the Step Control Table and records them.

4. SIO's and EXCP's

The number of Start Input/Output instructions (SIO's) and Execute Channel Program SVC's (EXCP's) recorded by the accounting routine are measures of the I/O activity of a job step. EXCP's are requests for the EXCP Supervisor to perform some I/O function. SIO's are issued by the EXCP Supervisor and are the actual operations required to perform the requested I/O. Each SIO and EXCP is counted when it is recorded in the Trace Table. The totals are kept in work counters in the work area.

5. WAIT Count

A total is kept of the number of times the whole job step is in the Wait state. This is a count of the intervals whose durations are added together to calculate Wait Time. The total is kept in a one-word counter in the work area.

6. Actual Amount of Main Core and Large Capacity Storage

The Core and LCS used are calculated by subtracting the minimum free core or LCS in a requested region from the region size requested for a job step. An example is shown in Fig. 2. In this example, core used = 200K - 50K = 150K.

This statistic usually makes a good value to use for region size the next time the job is run. However, core fragmentation can present a problem (Fig. 3). Although the core used in Fig. 3 is 200K - 50K = 150K, the fragmentation necessitates a region request of 200K to run the job step. The value 150K is still useful even in this case: It can be used by the programmer as the smallest region size possible, if he eliminates the fragmentation.

More specifically, the amount of storage used is calculated by adding up all the Free Block Queue Elements for the Partition Queue Element each time a Getmain is issued. This sum is compared to the current value in either the Actual CPU Used field or the Actual LCS Used field of the work area. If the new sum is smaller than the value in the work area, the new sum replaces the work area value. These work area fields are initialized at step initiation with the number of bytes of storage requested in each of the two kinds of storage. The requested region sizes in K are stored by OS in the Step Control Table, and these values are multiplied by 1024 to get the number of bytes requested.

7. Recommended Time Estimate

The time estimate in message ACCT001 is calculated by subtracting the time remaining for the job in the Timer Queue Element from the time estimate used on the job card, adding 10 seconds, and rounding to the next higher minute. It is the recommended time estimate for the job. This is the value that should be used on the job card the next time the job is run. Refer to Section V for more information on time estimates.

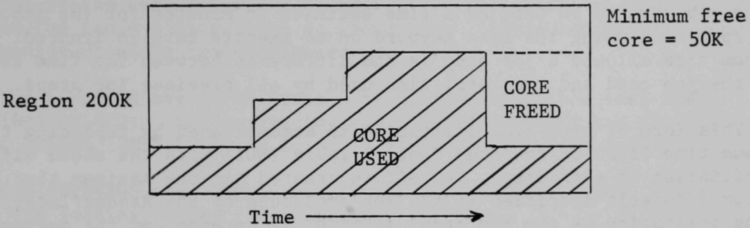


Fig. 2. Determination of core usage without core fragmentation

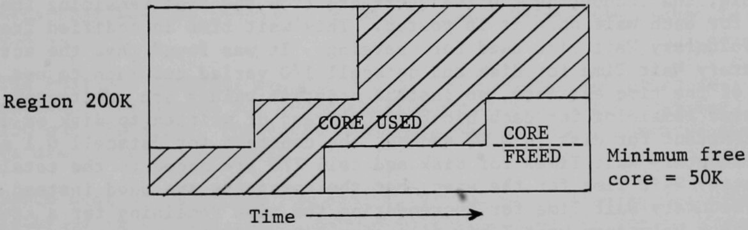


Fig. 3. Determination of core usage with core fragmentation

V. TIME ESTIMATES

In addition to its primary function of calculating statistics, the accounting routine causes the fourth field of the accounting information on the job card to be used as a time estimate in minutes for the job. Time specified using the time keyword on an execute card is ignored. The maximum time allowed a job step is the difference between the time estimate from the job card and the total time used by all previous job steps.

This form of step-time limitation is accomplished by replacing the maximum time field in the Step Control Table (SCT) with the above difference at initiation of each step. The SCT is created and the maximum time filled in from a default specified in the RDR procedure by the Reader/Interpreter. During initiation of the job step, IEFACTRT is entered and the maximum time in the SCT is modified. The last initiator module to get control before the job step is executed is IEFSD263 (the initiator attach routine). It issues the STIMER macro using the maximum time in the SCT, and then attaches the job step. A default job time of three minutes is currently supplied for a job with a blank or omitted time field on the job card.

OS issues the STIMER with the TASK option. This causes the time remaining to be decremented only when the step is using the CPU. When the value reaches zero the job is cancelled for time exceeded. In addition to this, the accounting routine subtracts from the time remaining the wait time for each wait when it is posted. This wait time is modified from the Voluntary Wait Time used for charging. It was found that the actual Voluntary Wait Time for disk and datacell I/O varied too much to use as part of the time estimate, so instead constant values are subtracted from the time remaining for each block (EXCP) read or written to disk or datacell. The constant for disk is 0.03 second, the constant for datacell 0.1 second. The Voluntary Wait Times for disk and cell I/O are added to the total Voluntary Wait Time for the step, but the constants are used instead of the Voluntary Wait Time for decrementing the time remaining for a step. All other Voluntary Wait Times (i.e. for I/O to devices other than disk and cell) are subtracted from the time remaining.

VI. "TIME LEFT" ROUTINE

TLEFT is a FORTRAN and PL/1 callable subroutine. It returns to the calling program the time remaining for the job in hundredths of a second. The routine is called with one dummy argument. The value returned is in single precision, floating point, binary. The value of the argument remains unchanged.

Example: X = TLEFT(Y) causes X to be set equal to the 'TIME LEFT'

A listing of the routine appears in Appendix D.

VII. APPLYING THE ACCOUNTING ROUTINE TO THE SYSTEM

The following is a description of the steps necessary to implement this accounting routine. Some of these steps can be taken either before or after System Generation (in slightly different ways, of course), as long as the Stage 1 macros are as specified. This description covers implementation after System Generation.

The SCHEDULR macro for Stage 1 of System Generation must have ACCTRTN=SUPPLIED. This causes the dummy CSECT IEFACTRT and the System Output Writer (IEFWAD) to be included in the Initiator/Terminator.

The SUPRVSOR macro for State 1 must have TIMER=JOBSTEP. This will cause the jobstep timing function to be included.

The dummy CSECT IEFACTRT must be replaced with the accounting routine in both load modules in which it appears. (Refer to the listing of the System Generation Linkage Editor for Linkage Editor parameters and control cards; refer also to "Inserting an Accounting Routine into the Control Program" in Ref. [1].) A listing of the accounting routine that replaces the dummy CSECT appears in Appendix B.

The STAGE II macros IEAQGM and IEAQTR in SYS1.MODGEN must be updated, after which the Stage I assemblies that called these macros must be rerun. The cards that must be inserted in each of these macros are listed in Appendix C.

The POST SVC (IGC002) and the TTIMER SVC (IGC046) must both be reassembled. The source for these modules is in SYS1.CI535, members IEAQSY50 and IEAQST00. The cards that must be inserted in each of these modules are listed in Appendix C.

Space must be allocated for SYS1.ACCT (the DCB and space requirements are discussed in Section III). Reference should also be made to "Output from the Accounting Routine" in Ref. [2].

The four object modules described in the two paragraphs above must all be included in the relinkedit of the Nucleus. When these steps are completed, do an IPL and the accounting routine will be running.

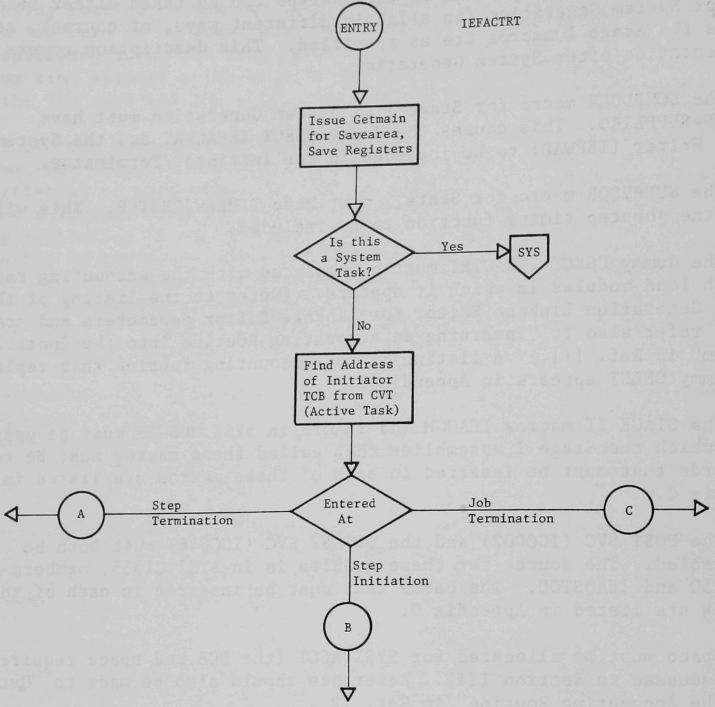
SYS1.ACCT can be output using the Utiltities. Scratching and reallocating the data set will reset the pointer that keeps track of which records to use next.

VIII. STAND-ALONE OS CONSIDERATIONS

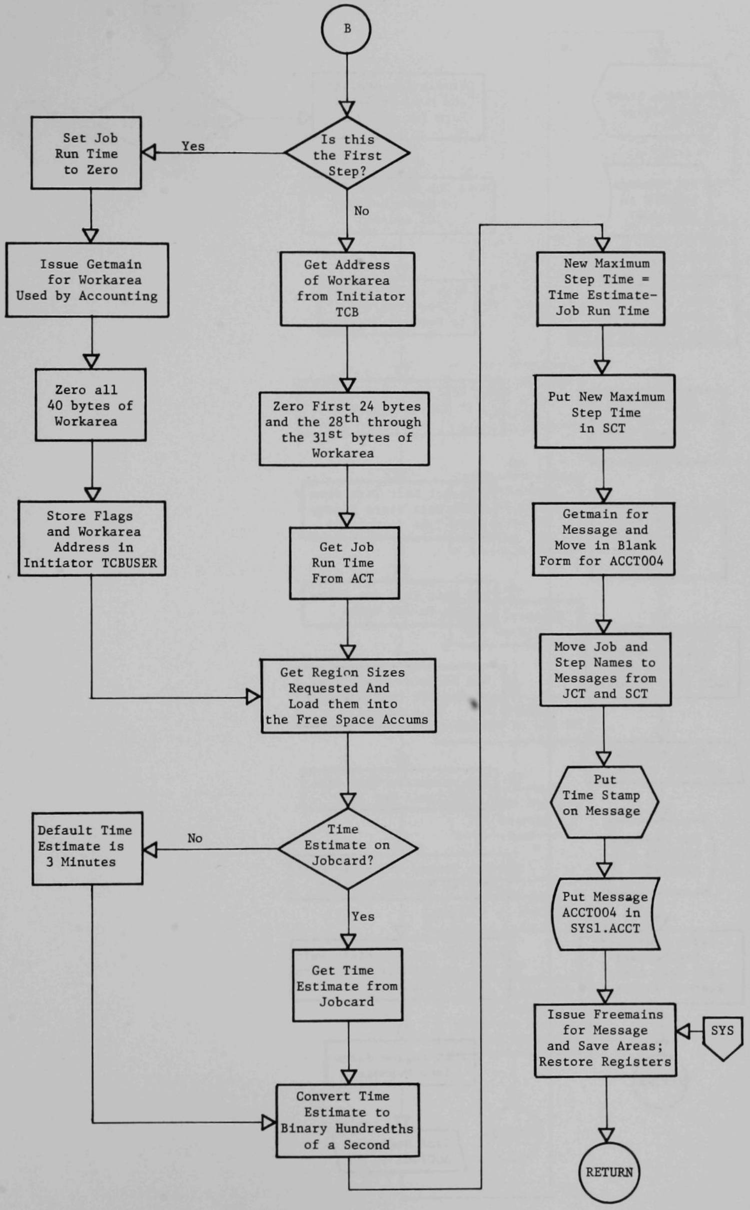
The Write to Operator commands used by the accounting routine to send messages to ASP would have to be changed on a stand-alone system and a System Output Writer would probably have to be written. An explanation of how to write such a routine appears under "Output from the Accounting Routine" in Ref. [1].

APPENDIX A

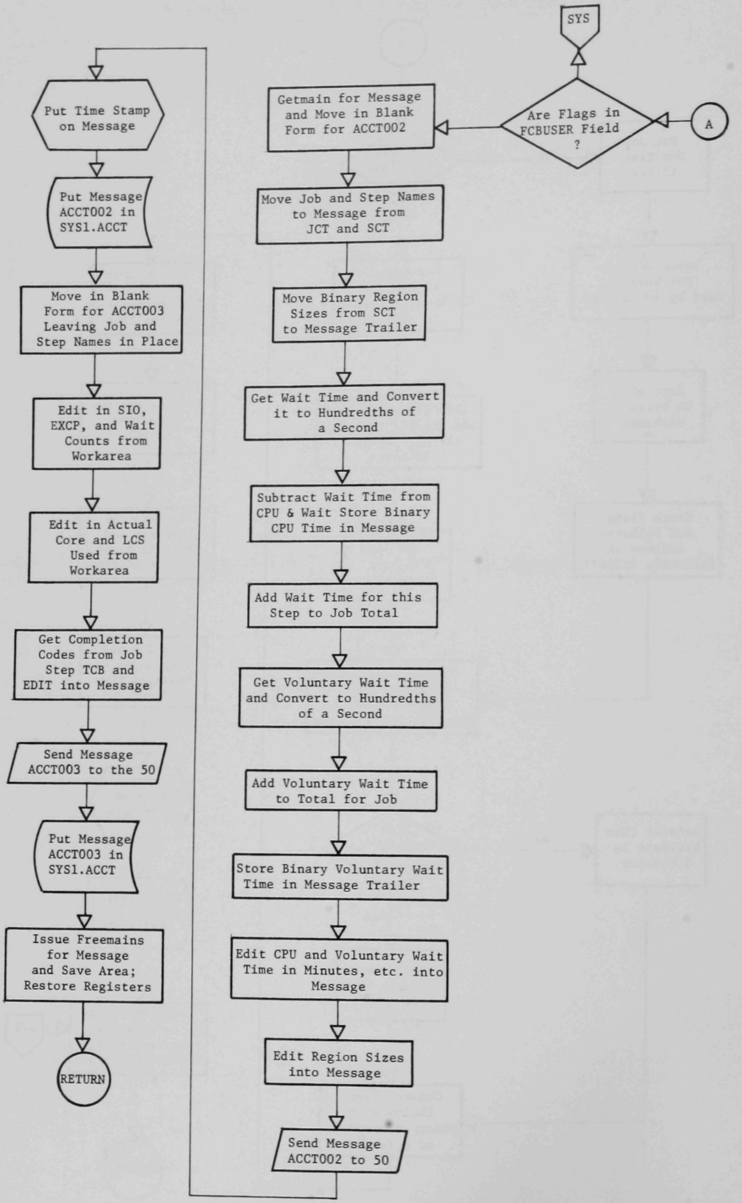
Flowchart of the Accounting Routine



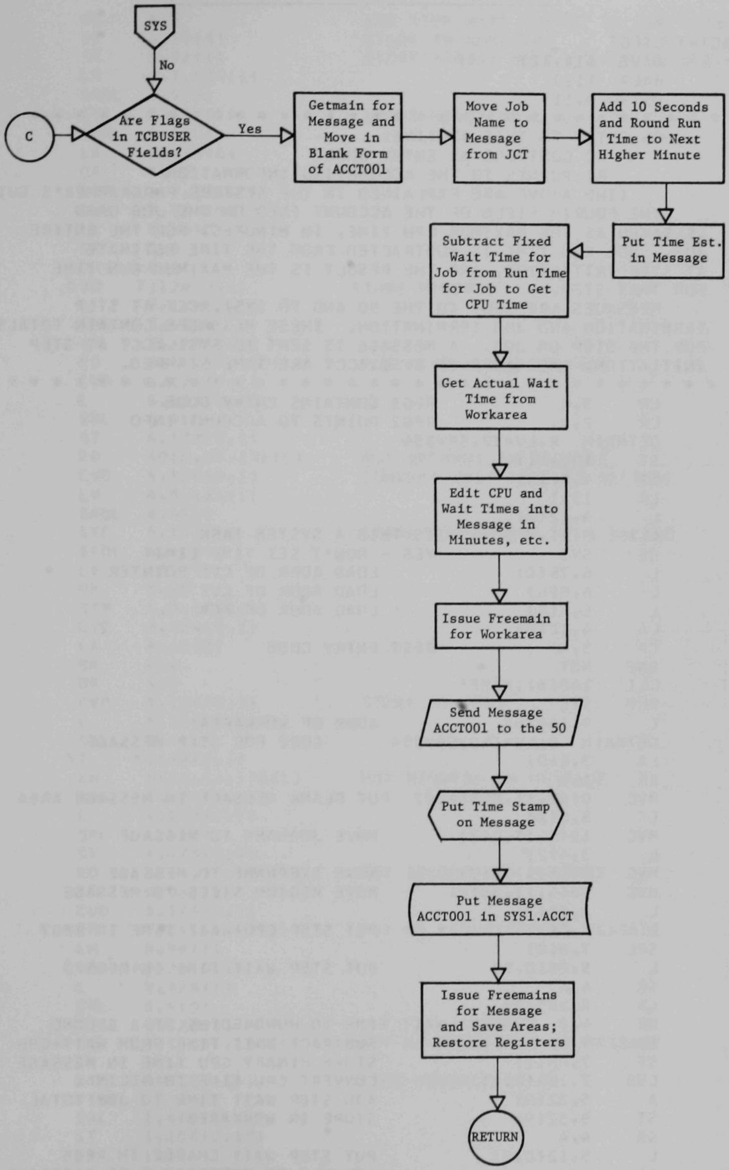
Flowchart of the Accounting Routine (Contd.)


B
Is this the First Step?
Yes
Set Job Run Time to Zero
Issue Getmain for Workarea Used by Accounting
Zero all 40 bytes of Workarea
Store Flags and Workarea Address in Initiator TCBUSER
No
Get Address of Workarea from Initiator TCB
Zero First 24 bytes and the 28th through the 31st bytes of Workarea
Get Job Run Time From ACT
Get Region Sizes Requested And Load them into the Free Space Accums
Time Estimate on Jobcard?
No
Default Time Estimate is 3 Minutes
Yes
Get Time Estimate from Jobcard
Convert Time Estimate to Binary Hundredths of a Second
New Maximum Step Time = Time Estimate- Job Run Time
Put New Maximum Step Time in SCT
Getmain for Message and Move in Blank Form for ACCT004
Move Job and Step Names to Messages from JCT and SCT
Put Time Stamp on Message
Put Message ACCT004 in SYS1.ACCT
Issue Freemains for Message and Save Areas; Restore Registers
SYS
RETURN

Flowchart of the Accounting Routine (Contd.)


SYS
Are Flags in FCBUSER Field ?
A
Getmain for Message and Move in Blank Form for ACCT002
Move Job and Step Names to Message from JCT and SCT
Move Binary Region Sizes from SCT to Message Trailer
Get Wait Time and Convert it to Hundredths of a Second
Subtract Wait Time from CPU & Wait Store Binary CPU Time in Message
Add Wait Time for this Step to Job Total
Get Voluntary Wait Time and Convert to Hundredths of a Second
Add Voluntary Wait Time to Total for Job
Store Binary Voluntary Wait Time in Message Trailer
Edit CPU and Voluntary Wait Time in Minutes, etc. into Message
Edit Region Sizes into Message
Send Message ACCT002 to 50
Put Time Stamp on Message
Put Message ACCT002 in SYS1.ACCT
Move in Blank Form for ACCT003 Leaving Job and Step Names in Place
Edit in SIO, EXCP, and Wait Counts from Workarea
Edit in Actual Core and LCS Used from Workarea
Get Completion Codes from Job Step TCB and EDIT into Message
Send Message ACCT003 to the 50
Put Message ACCT003 in SYS1.ACCT
Issue Freemains for Message and Save Area; Restore Registers
RETURN

Flowchart of the Accounting Routine (Contd.)


SYS
No
C
Are Flags in TCBUSER Fields?
Yes
Getmain for Message and Move in Blank Form of ACCT001
Move Job Name to Message from JCT
Add 10 Seconds and Round Run Time to Next Higher Minute
Put Time Est. in Message
Subtract Fixed Wait Time for Job from Run Time for Job to Get CPU Time
Get Actual Wait Time from Workarea
Edit CPU and Wait Times into Message in Minutes, etc.
Issue Freemain for Workarea
Send Message ACCT001 to the 50
Put Time Stamp on Message
Put Message ACCT001 in SYS1.ACCT
Issue Freemains for Message and Save Areas; Restore Registers
RETURN

APPENDIX B

Listing of the Accounting Routine

```
IEFACTRT CSECT
         SAVE  (14,12)
         BALR  11,0
         USING *,11
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*        ON ENTRY TO THIS ROUTINE:
*             R0 CONTAINS AN ENTRY CODE
*             R1 POINTS TO THE ACCOUNTING INFORMATION
*           (THE ABOVE ARE EXPLAINED IN THE SYSTEMS PROGRAMMER'S GUIDE)
*        THE FOURTH FIELD OF THE ACCOUNT INFO ON THE JOB CARD
*    IS TAKEN AS THE MAXIMUM CPU TIME, IN MINUTES, FOR THE ENTIRE
*    JOB.  JOB RUN TIME IS SUBTRACTED FROM THE TIME ESTIMATE
*    AT STEP INITIATION, AND THE RESULT IS THE MAXIMUM RUN TIME
*    FOR THAT STEP.
*        MESSAGES ARE SENT TO THE 50 AND TO SYS1.ACCT AT STEP
*    TERMINATION AND JOB TERMINATION.  THESE MESSAGES CONTAIN TOTALS
*    FOR THE STEP OR JOB.  A MESSAGE IS SENT TO SYS1.ACCT AT STEP
*    INITIATION.  MESSAGES IN SYS1.ACCT ARE TIME STAMPED.
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
         LR    3,0            REG3 CONTAINS ENTRY CODE
         LR    2,1            REG2 POINTS TO ACCOUNT INFO
         GETMAIN  R,LV=72,SP=254
         ST    13,4(1)
         ST    1,8(13)
         LR    13,1
         L     4,12(2)
         CLI   3(4),X'00'     IS THIS A SYSTEM TASK
         BE    SYS            YES - DON'T SET TIME LIMIT
         L     6,76(0)             LOAD ADDR OF CVT POINTER
         L     6,0(6)              LOAD ADDR OF CVT
         L     6,4(6)              LOAD ADDR OF TCB
         LA    4,12(0)
         CR    3,4            TEST ENTRY CODE
         BNE   NOT
         CLI   168(6),X'FF'
         BNE   SYS
         L     9,168(6)            ADDR OF WORKAREA
         GETMAIN  R,LV=140,SP=254       CORE FOR STEP MESSAGE
         LA    3,4(0)
         AR    1,3
         MVC   0(87,1),MESSAGE2  PUT BLANK MESSAGE IN MESSAGE AREA
         L     3,0(2)
         MVC   12(8,1),0(3)        MOVE JOBNAME TO MESSAGE
         L     3,4(2)
         MVC   21(8,1),0(3)        MOVE STEPNAME TO MESSAGE
         MVC   96(4,1),36(3)       MOVE REGION SIZES TO MESSAGE
         L     3,20(2)
         L     7,0(3)              PUT STEP CPU+WAIT TIME IN REG7
         SRL   7,8(0)
         L     5,28(0,9)           PUT STEP WAIT TIME IN REG5
         SR    4,4
         LA    8,385(0)
         DR    4,8  CONVERT WAIT TIME TO HUNDREDTHS OF A SECOND
         SR    7,5                 SUBTRACT WAIT TIME FROM WAIT+CPU
         ST    7,88(1)             STORE BINARY CPU TIME IN MESSAGE
         CVD   7,124(0,1)          CONVERT CPU TIME TO DECIMAL
         A     5,32(9)             ADD STEP WAIT TIME TO JOB TOTAL
         ST    5,32(9)             STORE IN WORKAREA
         SR    4,4
         L     5,12(0,9)           PUT STEP WAIT CHARGE IN REG5
         DR    4,8                 CONVERT TO HUNDREDTHS OF A SECOND
         L     6,24(9)             GET JOB TOTAL WAIT CHARGE
```

Listing of the Accounting Routine (Contd.)

```
 AR     6,5                    ADD STEP WAIT CHARGE TO JOB TOTAL
 ST     6,24(9)                STORE IN WORKAREA
 ST     5,92(1)                STORE BINARY WAIT CHARGE IN MESSAGE
 LM     6,7,124(1)
 SRDL   6,4(0)
 STC    7,132(1)               STORE HUNDREDTHS OF A SECOND
 SRDL   6,4(0)
 LA     10,15(0)
 OR     7,10
 STM    6,7,124(1)
 CVB    7,124(0,1)
 LA     8,60(0)
 SR     6,6
 DR     6,8
CVD    7,124(0,1)             STORE MINUTES
 L      7,128(0,1)
 SRL    7,4(0)
 ST     7,128(0,1)
 ED     33(6,1),130(1)         PUT MINUTES IN MESSAGE
 CVD    6,124(0,1)
 L      6,128(0,1)
 SRL    6,4(0)
 ST     6,128(0,1)
 ED     40(6,1),131(1)         PUT SECONDS IN MESSAGE
 CVD    5,124(0,1)             CONVERT WAIT TIME TO DECIMAL
 LM     4,5,124(1)
 SRDL   4,4(0)
 STC    5,132(1)               STORE HUNDREDTHS OF A SECOND
 SRDL   4,4(0)
 LA     10,15(0)
 OR     5,10
 STM    4,5,124(1)
 CVB    5,124(0,1)
 LA     8,60(0)
 SR     4,4
 DR     4,8
 CVD    5,124(0,1)             STORE MINUTES
 L      5,128(0,1)
 SRL    5,4(0)
ST     5,128(0,1)
 ED     51(6,1),130(1)         PUT MINUTES IN MESSAGE
 CVD    4,124(0,1)
 L      4,128(0,1)
 SRL    4,4(0)
 ST     4,128(0,1)
 ED     58(6,1),131(1)         PUT SECONDS IN MESSAGE
 LH     8,96(1)
 CVD    8,124(0,1)
 ED     72(4,1),130(1)         PUT H0 REQUESTED IN MESSAGE
 LH     8,98(1)
CVD    8,124(0,1)
 L      8,128(1)
 SRL    8,4(0)
 ST     8,128(1)
 ED     80(5,1),130(1)         PUT H1 REQUESTED IN MESSAGE
 LR     10,1
 WTO    MF=(E,(1))             SEND MESSAGE TO 50
 TIME
 SRL    1,4(0)
 ST     1,104(0,10)
 LR     1,10
 ST     0,108(0,1)
 MVC    112(4,1),96(1)         SAVE RQSTED REGIONS
```

Listing of the Accounting Routine (Contd.)

```
          MVC    86(14,1),PATTERN
          ED     86(14,1),105(1)
          LA     6,4(0)
          SR     1,6
          LA     6,96
          STH    6,6(1)
          L      15,VCON
          BALR   14,15
          LA     3,4(0)
          AR     1,3
          MVC    0(11,1),MESSAGE3
          MVC    30(54,1),MESSAGE3+30  PUT BLANK MESSAGE IN AREA
          L      8,0(0,9)
          CVD    8,124(0,1)
          ED     33(6,1),129(1)       PUT SIO COUNT IN MESSAGE
          L      8,4(0,9)
          CVD    8,124(0,1)
          ED     44(6,1),129(1)       PUT EXCP COUNT IN MESSAGE
          L      8,8(0,9)
          CVD    8,124(0,1)
          ED     55(6,1),129(1)       PUT WAIT COUNT IN MESSAGE
          L      8,16(0,9)            LOAD NUMBER OF BYTES OF CORE USED
          SRL    8,10                 DIVIDE BY 1024
          LH     7,112(1)
          SR     7,8
          CVD    7,124(0,1)
          ED     72(4,1),130(1)       PUT ACTUAL CORE USED IN MESSAGE
          L      8,20(0,9)            LOAD NUMBER OF BYTES OF LCS USED
          SRL    8,10                 DIVIDE BY 1024
          LH     7,114(1)
          SR     7,8
          CVD    7,124(0,1)
          L      7,128(0,1)
          SLL    7,4(0)
          ST     7,128(0,1)
          ED     80(5,1),129(1)       PUT ACTUAL LCS USED IN MESSAGE
          LR     10,1
          WTO    MF=(E,(1))                 SEND MESSAGE TO 50
          LR     1,10
          MVC    84(2,1),PATTERN
          LA     6,4(0)
          SR     1,6
          LA     6,96
          STH    6,6(1)
          L      15,VCON
          BALR   14,15
          LR     1,10
          LA     3,4(0)
          SR     1,3
          FREEMAIN  R,LV=140,A=(1),SP=254
          LR     1,13
          L      13,4(1)
          FREEMAIN  R,LV=72,A=(1),SP=254
          RETURN (14,12)        IF SO RETURN
NOT       BH     WTO            ENTERED AT JOB TERMINATION
          L      3,28(2)
          CLI    1(3),X'01'     IS THIS JOB INITIATION
          BE     JOB            THIS IS JOB INITIATION
          L      1,168(6)           LOAD ADDR OF WORKAREA
          MVC    0(24,1),WORKAREA   INITIALIZE WORKAREA
          MVC    28(4,1),WORKAREA
          L      3,12(2)
          L      9,0(0,3)
```

Listing of the Accounting Routine (Contd.)

```
         SRL   9,8(0)        LOAD JOB RUN TIME INTO REG9
         B     JOBCARD
JOB      SR    9,9           JOB RUN TIME IS ZERO
         GETMAIN  R,LV=40,SP=255
         MVC   0(40,1),WORKAREA   INITIALIZE WORKAREA
         O     1,MARKER
         ST    1,168(6)           STORE ADDR OF WORKAREA IN TCB
         L     3,12(2)
JOBCARD  L     5,20(12)           FIND SCT FROM LCT
         LH    4,88(5)            LOAD CPU REGION SIZE FROM SCT
         SLL   4,10               MULTIPLY BY 1024
         ST    4,16(0,1)          STORE REGION SIZE IN FREE SPACE ACCUM
         LH    4,90(5)            LOAD LCS REGION SIZE FROM SCT
         SLL   4,10               MULTIPLY BY 1024
         ST    4,20(0,1)          STORE REGION SIZE IN FREE SPACE ACCUM
         L     8,4(0,10)
         SRL   8,24(0)
         SLL   8,24(0)       GET INTER STEP STATUS INDICATORS
         CLI   3(3),X'04'    ARE THERE 4 ACCOUNT FIELDS
         BNE   DEFALT        NO - DEFAULT TO 3 MINUTES
         L     3,16(2)
         SR    4,4
         SR    5,5
         LA    5,1(0)
         IC    4,0(0,3)
         ALR   4,5
         ALR   3,4
         IC    4,0(0,3)
         ALR   4,5
         ALR   3,4
         IC    4,0(0,3)
         ALR   4,5
         ALR   3,4           REG3 NOW POINTS TO C4
         CLI   0(3),X'00'    IS THE TIME FIELD BLANK
         BE    DEFALT        YES - DEFAULT TO 3 MINUTES
         SR    6,6
         IC    4,C(0,3)      PUT TIME FIELD BYTE COUNT IN REG4
         SR    7,7
SHIFT    ALR   3,5           *                                       *
         SLDL  6,8(0)        THIS LOOP TAKES TIME ESTIMATE
         IC    7,0(0,3)      FROM JOB CARD AND PUTS IT IN REG6
         BCT   4,SHIFT       *                                       *
         GETMAIN  R,LV=16,SP=254
         STM   6,7,0(1)
         PACK  8(8,1),0(8,1)
         CVB   7,8(0,1)      CONVERT TIME EST TO BINARY
         FREEMAIN  R,LV=16,A=(1),SP=254
FIRST    LA    5,60(0)
         MR    6,5
         LA    5,100(0)
         MR    6,5
         AR    8,7
*
*        THE TIME EST IN HUNDRETHS OF A SECOND AND THE INTER STEP
*         INDICATORS NOW OCCUPY REG8
*
         SR    8,9           MAXTIME=TIME ESTIMATE - JOB RUN TIME
         BNL   TIMELEFT
         SR    8,8
TIMELEFT L     3,4(2)
         LA    4,48(0)
         SR    3,4           REG3 NOW POINTS TO THE SCT+4
         ST    8,C(0,3)      ORIGINAL MAXTIME TO SCT
```

Listing of the Accounting Routine (Contd.)

```
*
*     CODE TO PUT OUT STEP INITIATION MESSAGE
*
         GETMAIN  R,LV=140,SP=254
         LA     4,4(0)
         AR     1,4
         MVC    0(87,1),MESSAGE4
         L      3,0(2)
         MVC    12(8,1),0(3)          MOVE JOBNAME TO MESSAGE
         L      3,4(2)
         MVC    21(8,1),0(3)          MOVE STEPNAME TO MESSAGE
         LR     10,1
         TIME
         MVC    86(14,10),PATTERN
         SRL    1,4(0)
         ST     1,104(0,10)
         ST     0,108(0,10)
         ED     86(14,10),105(10)
         LR     1,10
         LA     6,4(0)
         SR     1,6
         LA     6,96
         STH    6,6(1)
         LR     6,1
         L      15,VCON
         BALR   14,15
         LR     1,6
         FREEMAIN  R,LV=140,A=(1),SP=254
SYS      LR     1,13
         L      13,4(1)
         FREEMAIN  R,LV=72,A=(1),SP=254
         RETURN (14,12)
DEFALT   LA     7,3(0)          DEFAULT TIME EST IS 3 MINUTES
         B      FIRST
*
*     SEND JOBNAME AND CPU TIME TO 50 VIA WTO MESSAGE
*
WTO      L      3,0(2)
         CLI    168(6),X'FF'
         BNE    SYS
         L      4,168(6)
         L      10,32(0,4)
         GETMAIN  R,LV=104,SP=254
         LA     6,4(0)
         AR     1,6
         MVC    0(87,1),MESSAGE1      PUT BLANK MESSAGE IN WORK AREA
         MVC    12(8,1),0(3)          MOVE JOBNAME TO MESSAGE
         L      3,12(0,2)
         L      8,0(0,3)
         SRL    8,8(0)          PUT JOB RUN TIME IN REG8
         SR     8,10
         SR     6,6
         LR     7,8
         LA     9,100(0)
         DR     6,9                   GET CPU+WAIT IN SECONDS
         SR     6,6
         LA     9,70(0)
         AR     7,9
         LA     9,60(0)
         DR     6,9                   GET CPU+WAIT IN MINUTES
         CVD    7,84(0,1)
         L      7,88(0,1)
         SRL    7,4(0)
```

Listing of the Accounting Routine (Contd.)

```
 ST     7,88(0,1)
 ED     76(5,1),90(1)         PUT TIME ESTIMATE IN MESSAGE
 SR     8,10
 CVD    8,84(0,1)             CONVERT RUN TIME TO DECIMAL
 LM     6,7,84(1)
 SRDL   6,4(0)
 STC    7,92(1)               STORE HUNDREDTHS OF A SECOND
 SRDL   6,4(0)
 LA     9,15(0)
 OR     7,9
 STM    6,7,84(1)
 CVB    7,84(0,1)
 LA     8,60(0)
 SR     6,6
 DR     6,8
 CVD    7,84(0,1)             STORE MINUTES
 L      7,88(0,1)
 SRL    7,4(0)
 ST     7,88(0,1)
 ED     30(6,1),90(1)         PUT MINUTES IN MESSAGE
 CVD    6,84(0,1)
 L      6,88(0,1)
 SRL    6,4(0)
 ST     6,88(0,1)
 ED     37(6,1),91(1)         PUT SECONDS IN MESSAGE
 L      8,24(0,4)
 CVD    8,84(0,1)             CONVERT RUN TIME TO DECIMAL
 LM     6,7,84(1)
 SRDL   6,4(0)
 STC    7,92(1)               STORE HUNDREDTHS OF A SECOND
 SRDL   6,4(0)
 LA     9,15(0)
 OR     7,9
 STM    6,7,84(1)
 CVB    7,84(0,1)
 LA     8,60(0)
 SR     6,6
 DR     6,8
 CVD    7,84(0,1)             STORE MINUTES
 L      7,88(0,1)
 SRL    7,4(0)
 ST     7,88(0,1)
 ED     54(6,1),90(1)         PUT MINUTES IN MESSAGE
 CVD    6,84(0,1)
 L      6,88(0,1)
 SRL    6,4(0)
 ST     6,88(0,1)
 ED     61(6,1),91(1)         PUT SECONDS IN MESSAGE
 LR     9,1
 WTO    MF=(E,(1))      SEND MESSAGE TO 50
 TIME
SRL    1,4(0)
 ST     1,100(0,9)
 LR     1,9
 ST     0,104(0,1)
 MVC    86(14,1),PATTERN
 ED     86(14,1),101(1)
 LA     6,4(0)
 SR     1,6
 LA     6,96
 STH    6,6(1)
 L      15,VCON
 BALR   14,15
```

Listing of the Accounting Routine (Contd.)

```
         LR    1,4
         FREEMAIN  R,LV=40,A=(1),SP=255
         LR    1,9
         LA    6,4(0)
         SR    1,6
         FREEMAIN  R,LV=108,A=(1),SP=254
         LR    1,13
         L     13,4(1)
         FREEMAIN  R,LV=72,A=(1),SP=254
         RETURN  (14,12)
WORKAREA DS    0F
         DC    14F'0'
MARKER   DC    XL4'FF000000'
UNITS    DC    XL4'0023366C'
PATTERN  DC    CL14'
BLANKS  DC    CL5'
MESSAGE1 WTO   'ACCT001         TOTAL CPU      .   .  TOTAL WAIT      X
               .    .   TIME EST    ',MF=L
MESSAGE2 WTO   'ACCT002        /         CPU      .   .   WAIT        . X
                 .   RQST HC    K,H1     K CCCCWWWW0011',MF=L
MESSAGE3 WTO   'ACCT003        /         SIO       EXCP        WAIT     X
                  USED HO     K,H1      K',MF=L
MESSAGE4 WTO   'ACCT004        /           STEP INITIATED               X
                                           ',MF=L
VCON     DC    V(IEFWAD)
         END
```

APPENDIX C

Listings of OS Modifications

Update the Macro IEAQTR with these cards.

```
          L      11,TCB(TRPTRG)                                            ANLACT 06380516
          L      11,124(11)          LOAD ADDR OF JOB STEP TCB             ANLACT 06381016
          L      11,132(11)          LOAD ADDR OF INIT TCB                 ANLACT 06381516
          CLI    168(11),X'FF'       IS THIS A USER TCB                    ANLACT 06382016
          BNE    *+20                NO - IT IS A SYSTEM TCB               ANLACT 06382516
          L      11,168(11)          LOAD ADDR OF WORKAREA                 ANLACT 06383016
          LA     12,1                                                      ANLACT 06383516
          A      12,0(11)            ADD ONE TO SIO COUNT                  ANLACT 06384016
          ST     12,0(11)            STORE NEW SIO COUNT IN WORKAREAANLACT 06384516
          L      13,4(14)  LOAD ADDR OF CURRENT TCB                        ANLACT 07490516
          L      13,124(13)  LOAD ADDR OF JOB STEP TCB                     ANLACT 07491016
          L      13,132(13)  LOAD ADDR OF INIT TCB                         ANLACT 07491516
          CLI    168(13),X'FF'  IS THIS A USER TCB                         ANLACT 07492016
          BNE    POSTCOM             NO - THIS IS A SYSTEM TCB             ANLACT 07492516
          LH     11,2(TRPTRG)        LOAD SVC IDENT AND NUMBER             ANLACT 07493016
          C      11,EXCP             IS THIS A SVC 0                       ANLACT 07493516
          BNE    POSTCOM             NO - BRANCH AROUND                    ANLACT 07494016
          L      14,168(13)          LOAD ADDR OF WORKAREA                 ANLACT 07494517
          LA     11,1                                                      ANLACT 07495016
          A      11,4(14)            ADD OME TO EXCP COUNT                 ANLACT 07496217
          ST     11,4(14)            STORE NEW EXCP COUNT                  ANLACT 07496317
          L      11,16(TRPTRG)       LOAD IOB ADDR                         ANLACT 07496417
          L      11,20(11)           LOAD DCB ADDR                         ANLACT 07496517
          L      11,44(11)           LOAD DEB ADDR                         ANLACT 07496617
          L      11,32(11)           LOAD UCB ADDR                         ANLACT 07496717
          CLI    18(11),X'20'        IS THIS DIRECT ACCESS                 ANLACT 07496817
          BNE    POSTCOM             BRANCH IF NOT                         ANLACT 07496917
          CLI    19(11),X'08'        IS THIS A 2314                        ANLACT 07497017
          BE     DISK                BRANCH IF IT IS                       ANLACT 07497117
          CLI    19(11),X'05'        IS THIS A 2321                        ANLACT 07497217
          BE     CELL                IF IT IS BRANCH                       ANLACT 07497317
          BC     15,POSTCOM             TO CLEAN UP                               07500000
DISK      LH     11,38(14)           LOAD PENDING DISK EXCP COUNT          ANLACT 07500517
          C      11,EXCPMAX          IS COUNTER FULL?                      ANLACT 07501017
          BE     POSTCOM             YES.  DO NOT INCREMENT                ANLACT 07501517
          LA     12,1(0)                                                   ANLACT 07502017
          AR     11,12               ADD ONE TO COUNT                      ANLACT 07502517
          STH    11,38(14)           STORE NEW TOTAL                       ANLACT 07503017
          B      POSTCOM                                                   ANLACT 07503517
CELL      LH     11,36(14)           LOAD PENDING CELL EXCP COUNT          ANLACT 07504017
          C      11,EXCPMAX          IS COUNTER FULL?                      ANLACT 07504517
          BE POSTCOM                 YES.  DO NOT INCREMENT                ANLACT 07505017
          LA     12,1(0)                                                   ANLACT 07505517
          AR     11,12               ADD ONE TO OUNT                       ANLACT 07506017
          STH    11,36(14)           STORE NEW TOTAL                       ANLACT 07507017
          B      POSTCOM                                                   ANLACT 07507517
EXCP      DC     F'8192'                                                   ANLACT 07509817
EXCPMAX   DC     X'0000EFFF'                                               ANLACT 07509917
```

Update the Macro IEAQGM with these cards.

```
GDQEBLD   STM    10,15,SAVEDRS       SAVE REGISTERS                        ANLACT 43531016
          LR     11,RTCB             LOAD ADDR OF CURRENT TCB              ANLACT 43531516
          L      11,124(11)          LOAD ADDR OF JOB STEP TCB             ANLACT 43532016
          L      11,132(11)          LOAD ADDR OF INIT TCB                 ANLACT 43532516
          CLI    168(11),X'FF'       IS THIS A SYSTEM TCB                  ANLACT 43533316
          BNF    LASTPQE+10          YES - BRANCH AROUND                   ANLACT 43533516
          L      14,TCBPQE(11)       LOAD ADDR OF DUMMY PQE-8              ANLACT 43534016
          L      11,168(11)          LOAD ADDR OF WORKAREA                 ANLACT 43543016
```

Listings of OS Modifications (Contd.)

```
LOOPPQE  L     14,8(14)          LOAD ADDR OF FIRST PQE ON CHAINANLACT 43534516
         LA    14,0(14)                                        ANLACT 43535016
         SR    13,13                                           ANLACT 43535516
         L     15,0(14)          LOAD ADDR OF FIRST FBQE ON CHAIANLACT 43536016
         LA    15,0(15)                                        ANLACT 43536516
COUNT    CR    14,15             ARE THERE ANY MORE FBQE'S     ANLACT 43537016
         BE    ALLGONE                                         ANLACT 43537516
         L     12,8(15)          LOAD SIZE OF FREE BLOCK       ANLACT 43538016
         LA    12,0(12)          ZERO HIGH ORDER BYTE          ANLACT 43538516
         AR    13,12             ADD SIZE OF BLOCK TO TOAL     ANLACT 43539016
         L     15,0(15)                                        ANLACT 43539216
         LA    15,0(15)          LOAD ADDR OF NEXT FBQE        ANLACT 43540016
         B     COUNT                                           ANLACT 43540516
SAVEDRS  DC    6F'0'                                           ANLACT 43541016
ALLGONE  SR    12,12                                           ANLACT 43541516
         IC    12,29(14)         INSERT PQEHRID                ANLACT 43542016
         SLL   12,2              MULTIPLY BY FOUR              ANLACT 43542516
         C     13,16(12,11)      IS THIS A NEW MINIMUM FREE AREAANLACT 43543516
         BNL   LASTPQE           NO NEW MINIMUM                ANLACT 43544016
         ST    13,16(12,11)      STORE NEW MINIMUM FREE AREA   ANLACT 43544516
LASTPQE  SR    13,13                                           ANLACT 43545016
         C     13,8(14)           IS THIS THE LAST PQE ON THE CHANLACT 43545516
         BNE   LOOPPQE           NO - LOOP BACK                ANLACT 43546016
         LM    10,15,SAVEDRS     RESTORE REGISTERS             ANLACT 43546516
         LR    RBLOCK1,RBLOCK    SAVE BLOCK NUMBER AND SIZE    ANLACT 43547016
*GDQEBLD LR    RBLOCK1,RBLOCK    SAVE BLOCK NUMBER AND SIZE           43560015
```

Update the Source Code for SVC46 with these cards.

```
         C     1,=X'0000000F'                                  ANLACT 02895117
         BNE   *+8                                             ANLACT 02895217
         LR    4,0                                             ANLACT 02895317
         SR    1,1                                             ANLACT 02895417
```

Update the Source Code for IEAQSY50 with these cards.

```
         L     6,168(XRWTCT1)      LOAD ADDR OF WORK AREA          ANLACT 17816616
         L     4,8(6)                                              ANLACT 17817101
         LA    5,1(0)                                              ANLACT 17817102
         AR    4,5                                                 ANLACT 17817103
         ST    4,8(6)                                              ANLACT 17817105
         L     5,W30MIN                                            ANLACT 17817112
         S     5,TQEVAL(XRWTCT1)         CALCULTATE LENGTH OF WAIT ANLACT 17817113
         L     4,12(6)                                             ANLACT 17817200
         AR    4,5                       CALCULATE NEW TOTAL WAIT  ANLACT 17817210
         ST    4,12(6)             STORE NEW TOTAL IN WORKAREA     ANLACT 17817220
        SR   3,3                                                   ANLACT 17817230
        SR   4,4                                                   ANLACT 17817240
        IC   4,59(0)             PUT UNIT ADDR IN REG4             ANLACT 17817250
         SRL   4,4(0)                                              ANLACT 17817260
        LA    3,3(0)                                               ANLACT 17817270
         CR    3,4                 WAS THIS WAITFOR DISK I/O       ANLACT 17817280
        BE    DISK               YES                               ANLACT 17817290
         LA    3,4(0)                                              ANLACT 17817300
        CR    3,4                WAS THIS WAIT FOR DISK I/O        ANLACT 17817310
        BE    DISK               YES                               ANLACT 17817320
         LA    3,9(0)                                              ANLACT 17817330
         CR    3,4                 WAS THIS WAIT FOR CELL I/O      ANLACT 17817340
         BE    DISK              YES                               ANLACT 17817350
DASC     SR    3,3                                                 ANLACT 17817360
         CH    3,38(6)              ARE ANY DISK EXCPS PENDING?    ANLACT 17817370
         BE    CELLEXCP            NO.  BRANCH                     ANLACT 17817380
         LH    2,38(6)             LOAD NUMBER OF PENDING DISK EXCANLACT 17817390
```

Listings of OS Modifications (Contd.)

```
         MH    2,DISKTIME          CALCULATE TIME CHARGE FOR EXCPSANLACT 17817400
         AR    5,2                                                ANLACT 17817410
CELLEXCP CH    3,36(6)             ARE ANY CELL EXCPS PENDING?    ANLACT 17817450
         BE    JOBTIME             NO                             ANLACT 17817460
         LR    3,5                                                ANLACT 17817470
         LH    5,36(6)             LOAC NUMBER OF PENDING CELL EXCANLACT 17817473
         MH    5,CELLTIME          CALCULATE TIME CHARGE          ANLACT 17817476
         AR    5,3                 TOTAL TIME CHARGE              ANLACT 17817479
JOBTIME  L     4,TQESAV(XRWTCT1)   LOAC OLD REMAINING JOB TIME    ANLACT 17817482
         SR    3,3                                                ANLACT 17817485
         ST    3,36(6)             ZERO PENDING EXCP COUNTERS     ANLACT 17817488
         SR    4,5                 SUBTRACT WAIT TIME FOR THIS    ANLACT 17817491
         BH    *+10                IS THERE TIME REMAINING        ANLACT 17817494
         AR    4,5                                                ANLACT 17817497
         LR    5,4                 THIS WAIT EQUALS TIME REMAININGANLACT 17817500
         SR    4,4                                                ANLACT 17817503
         ST    4,TQESAV(XRWTCT1)   STCRE NEW REMAININC TIME LEFT  ANLACT 17817506
         L     4,28(6)             GET STEP FIXED WAIT TIME       ANLACT 17817509
         AR    4,5                 ADD WAIT TIME FOR THIS WAIT    ANLACT 17817512
         ST    4,28(6)             STORE NEW STEP FIXED WAIT TIME ANLACT 17817515
DISK     SR    5,5                                                ANLACT 17871017
       B     CASC                                                 ANLACT 17872017
         DS    3H                                                 ANLACT 17873017
DISKTIME DC    X'0483'             .03 SECONDS IN TIMER UNITS     ANLACT 17874401
CELLTIME DC    X'0F0A'             .1 SECONDS IN TIMER UNITS      ANLACT 17875017
```

APPENDIX D

Listing of the "Time Left" Routine

```
*   THIS ROUTINE IS CALLABLE FROM FORTRAN OR PL1.  IT RETURNS THE
*   TIME LEFT BEFORE A JOB IS CANCELLED FOR TIME EXCEEDED.  TIME IS
*   MEASURED IN HUNDREDTHS OF SECONDS OF CPU PLUS VOLUNTARY WAIT TIME.
TLEFT    START
         B      10(15)          BRANCH AROUND CONSTANTS IN CALLING SEQ
         DC     X'5'
         DC     CL5'TLEFT'
         STM    14,12,12(13)
         BALR   12,0
         USING  *,12
         L      6,76(0)         LOAD ADDR OF CVT POINTER
         L      6,0(6)          LOAD ADDR OF CVT
         L      6,4(6)          LOAD ADDR OF ACTIVE TCB
         L      6,124(6)        LOAD ADDR OF JOB STEP TCB
         L      0,132(6)        LOAD ADDR OF INITIATOR TCB
         LR     6,1
         LA     1,15(0)
         SVC    46              ISSUE TTIMER SVC
*   TIME LEFT BEFORE JOB IS CANCELLED FOR TIME EXCEEDED IS RETURNED
*   IN REGISTER 0
         LA     2,385(0)
         LR     1,0
         SR     0,0
         DR     0,2             CONVERT TO HUNDREDTHS OF A SECOND
         LE     0,NORM
         AL     1,NORM
         ST     1,WORK
         AE     0,WORK
         SR     0,0
         C      0,0(6)
         BH     *+12
         L      6,4(6)
         STE    0,0(6)
         LM     2,12,28(13)     RESTORE REGISTERS
         MVI    12(13),X'FF'    INDICATE CONTROL RETURNED TO CALLER
         BCR    15,14
WORK     DS     F
NORM     DC     XL4'46000000'
         END
```

REFERENCES

1. *IBM System/360 Operating System Programmer's Guide* (C28-6550-5), pp. 39-47 (1968).

2. *IBM System/360 Operating System System Generation* (C28-6554-5), pp. 151-155 (1968).

x